DATA 100 Spring 2021 Final-Exam

INSTRUCTIONS

This is your exam. Complete it either at exam.cs61a.org or, if that doesn't work, by emailing course staff with your solutions before the exam deadline.

This exam is intended for the student with email address `<EMAILADDRESS>`. If this is not your email address, notify course staff immediately, as each exam is different. Do not distribute this exam PDF even after the exam ends, as some students may be taking the exam in a different time zone.

For questions with **circular bubbles**, you should select exactly *one* choice.

○ You must choose either this option

○ Or this one, but not both!

For questions with **square checkboxes**, you may select *multiple* choices.

□ You could select this choice.

□ You could select this one too!

You may start your exam now. Your exam is due at <DEADLINE> Pacific Time. Go to the next page to begin.

Preliminaries

You can complete and submit these questions before the exam starts.

(a) What is your full name?

(b) What is your Berkeley email?

(c) What is your student ID number?

(d) When are you taking this exam?

○ Wednesday 11:40am PDT

○ Wednesday 7:10pm PDT

○ Other

(e) Honor Code: *All work on this exam is my own.*

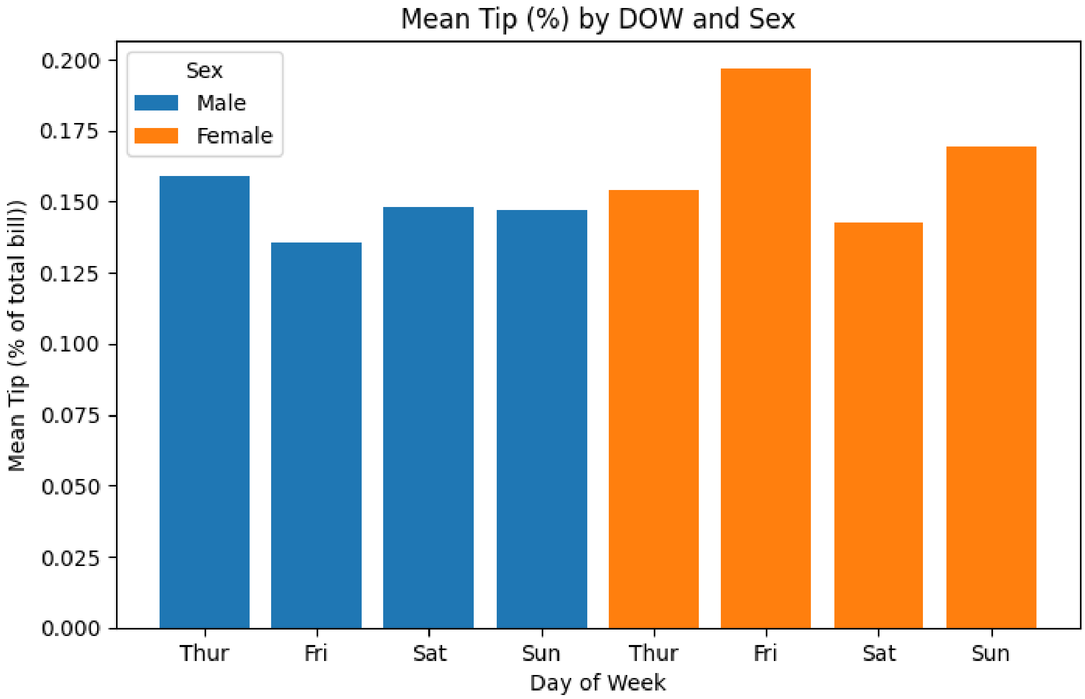
By writing your full name below, you are agreeing to this code:

(f) Important: You must copy the following statement exactly into the box below. Failure to do so may result in points deducted on the exam.

"I certify that all work on this exam is my own. I acknowledge that collaboration of any kind is forbidden, and that I will face severe penalties if I am caught, including at minimum, harsh penalties to my grade and a letter sent to the Center for Student Conduct."

1. **(7.0 points)**

   (a) **(2.0 pt)** Recall the tips dataset that we worked with on assignments in the past, which includes data about the tip on a restaurant bill as well as the day of week and the sex of the individual. The plot below attempts to examine patterns between the tip as a percentage of the bill and the sex of the individual by the day of week (DOW)



Select the best reason below for why the data visualization is misleading or poorly constructed.

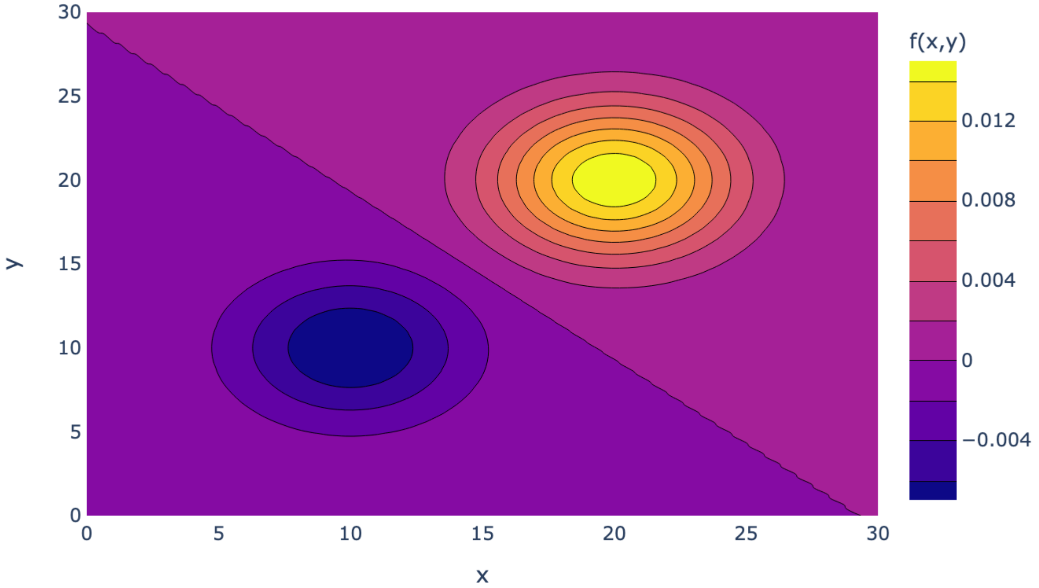
○ the y-axis should be log transformed

● the clustering of bars doesn't allow a key comparison to be made

○ the plot suffers from overplotting

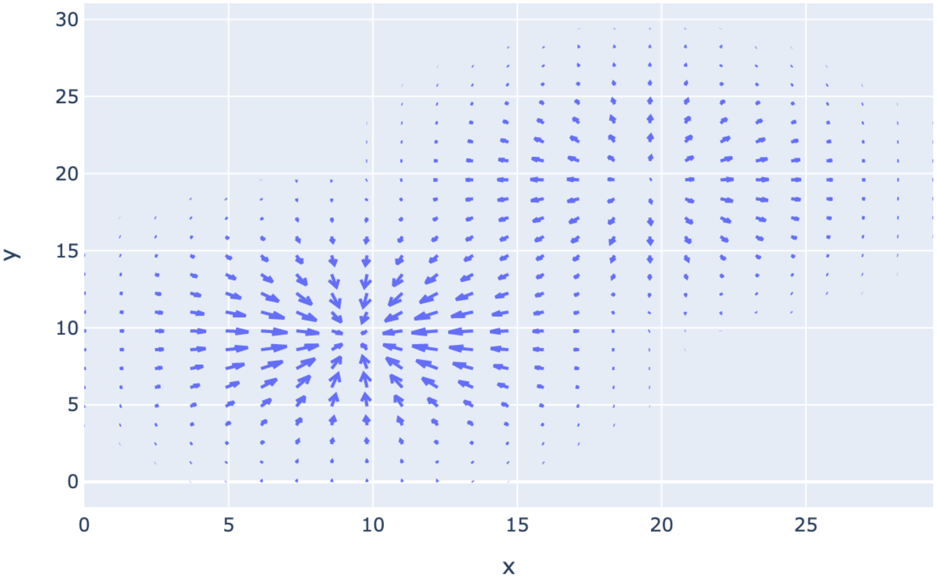
○ the bars for each day of week should be stacked on top of each other (e.g. the bar for "Thur" would have a total height of approximately 0.3)

(b) (2.0 pt) Consider the surface whose contour plot is provided below.

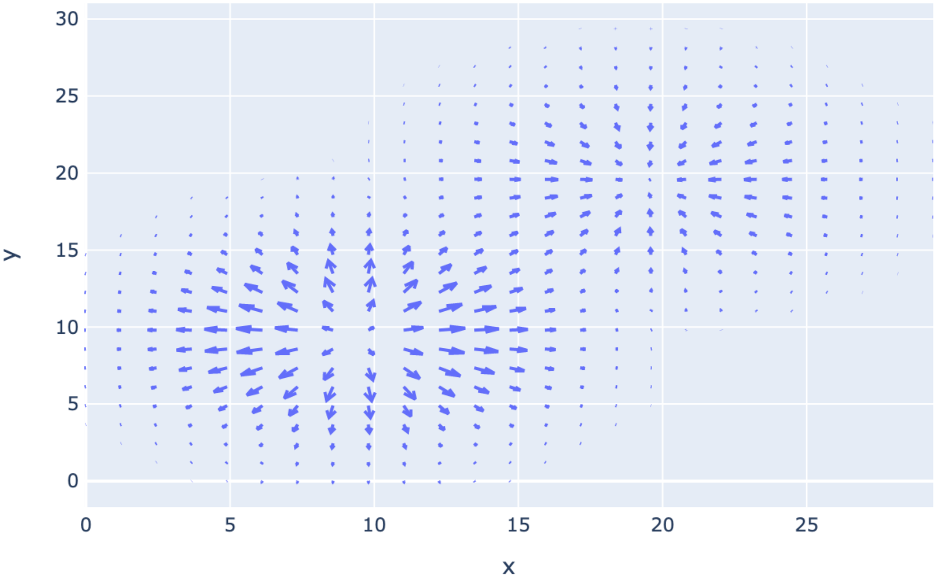


gradient fields most likely corresponds to the surface shown above?

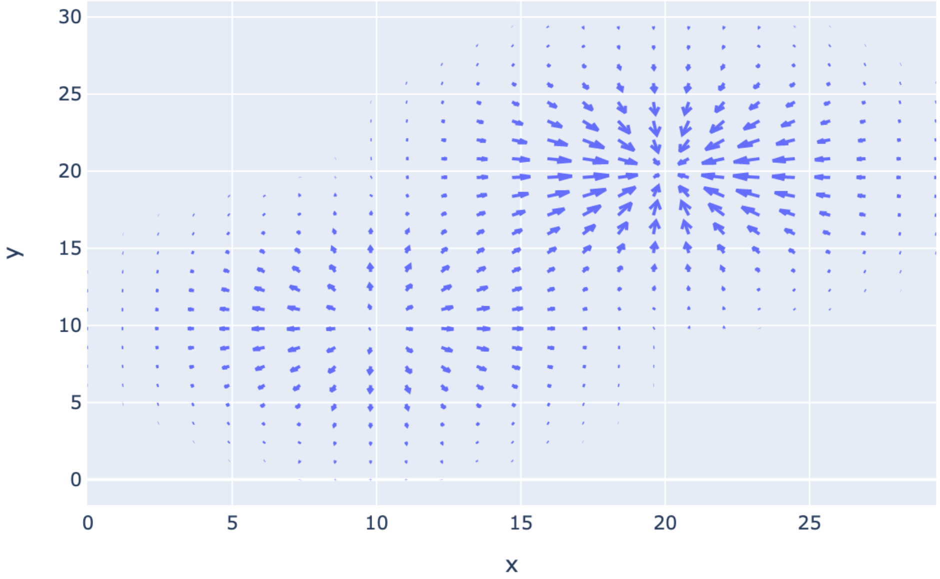
A **gradient field** is a plot that shows the direction and relative magnitude of the gradient of a surface on a 2-dimensional plot where each point has a vector pointing from it in the direction of the gradient at that point and the length of that vector is proportional to the magnitude of the gradient.


30
25
20
15
10
5
0
y
0
5
10
15
20
25
x

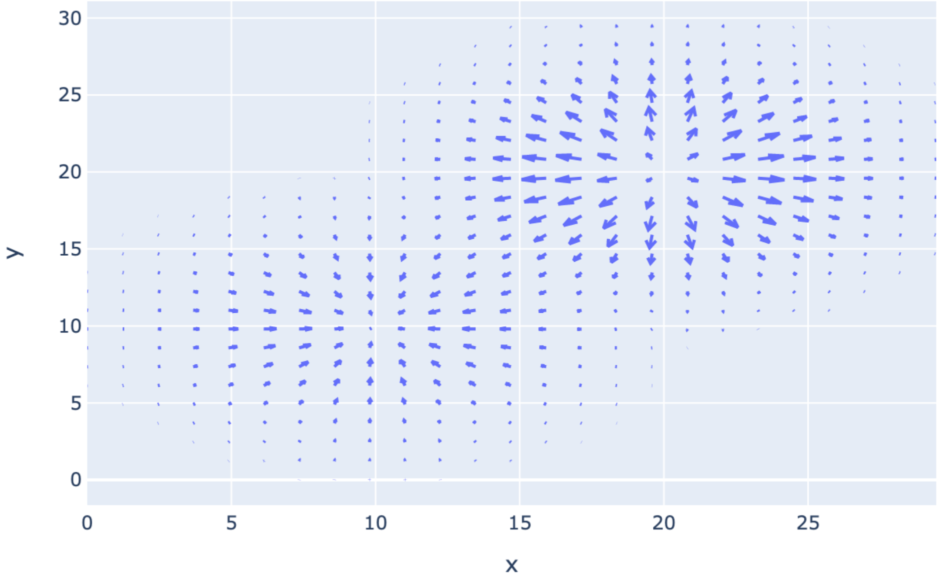
○


30
25
20
15
10
5
0
y
0
5
10
15
20
25
x

○



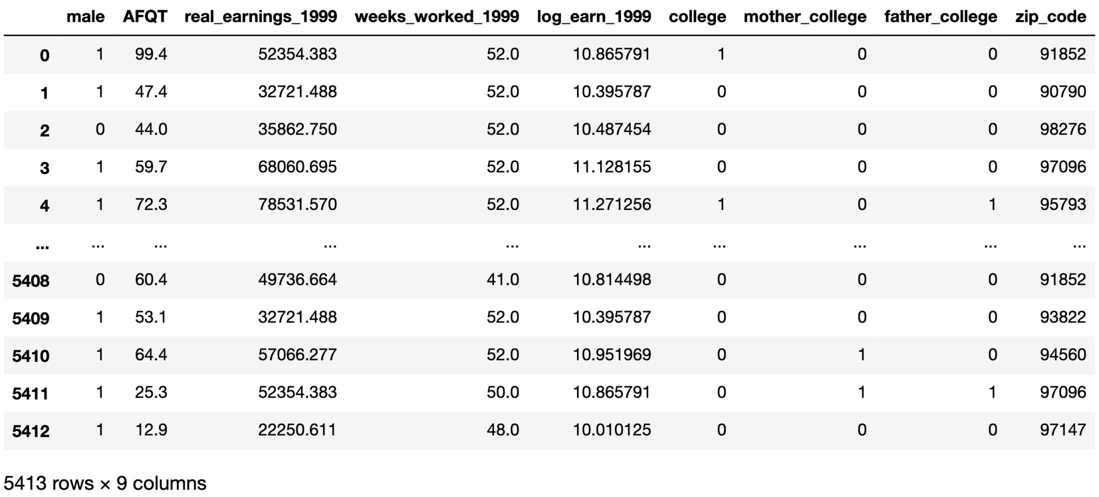
●



○

(c) (3.0 points)

We have read in some data as the dataframe `df`. Consider a subset of `df` below, which contains some information on the background of various individuals in the US.



	male	AFQT	real_earnings_1999	weeks_worked_1999	log_earn_1999	college	mother_college	father_college	zip_code
0	1	99.4	52354.383	52.0	10.865791	1	0	0	91852
1	1	47.4	32721.488	52.0	10.395787	0	0	0	90790
2	0	44.0	35862.750	52.0	10.487454	0	0	0	98276
3	1	59.7	68060.695	52.0	11.128155	0	0	0	97096
4	1	72.3	78531.570	52.0	11.271256	1	0	1	95793
...	...	...	...	...	...	...	...	...	...
5408	0	60.4	49736.664	41.0	10.814498	0	0	0	91852
5409	1	53.1	32721.488	52.0	10.395787	0	0	0	93822
5410	1	64.4	57066.277	52.0	10.951969	0	1	0	94560
5411	1	25.3	52354.383	50.0	10.865791	0	1	1	97096
5412	1	12.9	22250.611	48.0	10.010125	0	0	0	97147

5413 rows × 9 columns

i. (2.0 pt) Suppose we want to observe the relationship between and the distributions of the `AFQT` (an intelligence metric, with units percentile) and `log_earn_1999` (log of the individual's earnings in 1999) variables based on whether the individual's parents both went to college. Select the line of code below that generates the best plot to observe this relationship.

- A:

```
sns.kdeplot(x=df['AFQT'], y=df['log_earn_1999'],
hue=df['mother_college'] & df['father_college'])
```

- B:

```
sns.scatterplot(x=df['AFQT'], y=df['log_earn_1999'],
hue=df['mother_college'] & df['father_college'])
```

- C:

```
sns.lineplot(x=df['AFQT'], y=df['log_earn_1999'],
hue=df['mother_college'] & df['father_college'])
```

- D:

```
sns.kdeplot(x='AFQT', y='log_earn_1999', hue=['mother_college',
'father_college'], data=df)
```

- E:

```
sns.scatterplot(x='AFQT', y='log_earn_1999',hue=['mother_college',
'father_college'], data=df)
```

- F:

```
sns.lineplot(x='AFQT', y='log_earn_1999', hue=['mother_college',
'father_college'], data=df)
```

Hint: Consider overplotting.

● A

○ B

○ C

○ D

○ E

○ F

ii. **(1.0 pt)** Suppose we want to understand the relationship between `weeks_worked_1999` and the sex of the individual. We run the following code to generate a plot:

```
df2 = df.groupby("zip_code").mean().reset_index()
sns.lineplot("zip_code", "log_earn_1999", data=df2)
```

Select the reason below for why this plot would represent a bad data visualization.

● treats a categorical variable as a continuous variable

○ treats a continuous variable as a categorical variable

○ represents a density with a feature other than area

○ does not show the relationship between the variables of interest

2. (9.0 points)

(a) (4.0 points)

Recall that a random forest is created from a number of decision trees, with each decision tree created from a bootstrapped version of the original training set. One hyperparameter of a random forest is the number of decision trees we train to create the random forest.

Define T to be the number of decision trees used to create the random forest. Let's say we have two candidate values for T: $var1$ and $var2$. We want to perform $var3$ - fold cross-validation to determine the optimal value of T. Assume $var1$, $var2$, and $var3$ are integers.

i. (2.0 pt) In this cross-validation process, how many **random forests** will we train? Your answer should be in terms of $var1$, $var2$, and/or $var3$ and should be an integer.

var3

ii. (2.0 pt) In this cross-validation process, how many **decision trees** will we train? Your answer should be in terms of $var1$, $var2$, and/or $var3$ and should be an integer.

(var1 + var2) * var3

(b) **(2.0 pt)** Let's say we pick three hyperparameters to tune with cross-validation. We have 5 candidate values for hyperparameter 1, 6 candidate values for hyperparameter 2, and 7 candidate values for hyperparameter 3. We perform 4-fold cross validation to find the optimal combination of hyperparameters, across all possible combinations.

In this cross-validation process, how many **random forests** will we train? Your answer can be left as a product of multiple integers, e.g. "1 * 2 * 3", or simplified to a single integer, e.g. "6". (These are not the correct answers to the problem).

4 * 5 * 6 * 7 = 840

(c) **(3.0 pt)** Here is some code that attempts to implement the cross-validation procedure described above. However, it is buggy. In one sentence, describe the bug below.

You may assume the following:

- `X_train` is a `pd.DataFrame` that contains our design matrix, and `Y_train` is a `pd.Series` that contains our response variable, both for the full training set.
- Assume `ensemble.RandomForestClassifier(**args)` creates a random forest with the appropriate hyperparameter values. The bug is not on this line.
- The candidate values for each hyperparameter have been loaded into the lists `cands1`, `cands2`, and `cands3`, respectively.

```
from sklearn.model_selection import KFold
from sklearn import ensemble
import numpy as np
import pandas as pd

kf = KFold(n_splits = 4)

cv_scores = []
for cand1 in cands1:
    for cand2 in cands2:
        for cand3 in cands3:
            validation_accuracies = []
            for train_idx, valid_idx in kf.split(X_train):
                split_X_train, split_X_valid = X_train.iloc[train_idx], X_train.iloc[valid_idx]
                split_Y_train, split_Y_valid = Y_train.iloc[train_idx], Y_train.iloc[valid_idx]

                model = ensemble.RandomForestClassifier(**args)
                model.fit(X_train, Y_train)

                accuracy = np.mean(model.predict(split_X_valid) == split_Y_valid)
                validation_accuracies.append(accuracy)
            cv_scores.append(np.mean(validation_accuracies))
```

Each iteration of the algorithm trains a random forest on the entire training set, as opposed to the part of the training set that is not reserved for validation.

3. (14.0 points)

We are trying to train a decision tree for a classification task where 0 is the negative class and 1 is the positive class. We are given 8 data points each in pairs of (x_1, x_2) features.

(a) (3.0 pt)

x_1	x_2	y
3	4	1
2	1	0
1	3	1
5	9	0
9	6	1
7	2	1
4	7	0
8	8	1

What is the entropy at the root of the tree? Round to 4 decimal places.

$-(\frac{5}{8}log\frac{5}{8} + \frac{3}{8}log(\frac{3}{8})) = 0.6616$

(b) (2.0 pt) What is the gini inpurity at the root of the tree? Note that the formula for gini impurity is $1 - \sum_{i=1}^{c} p_i^2$ where p_i is the fraction of items labelled with class i and c is the total number of classes.

$1 - ((\frac{5}{8})^2 + (\frac{3}{8})^2) = 0.46875$

(c) **(4.0 pt)** Suppose we decide to split the root node with the rule $x_i \geq \beta$ where i = 1 or 2. Which of the following minimizes the weighted entropy of the two resulting child nodes.

● $x_1 \geq 6$

○ $x_1 \geq 3.5$

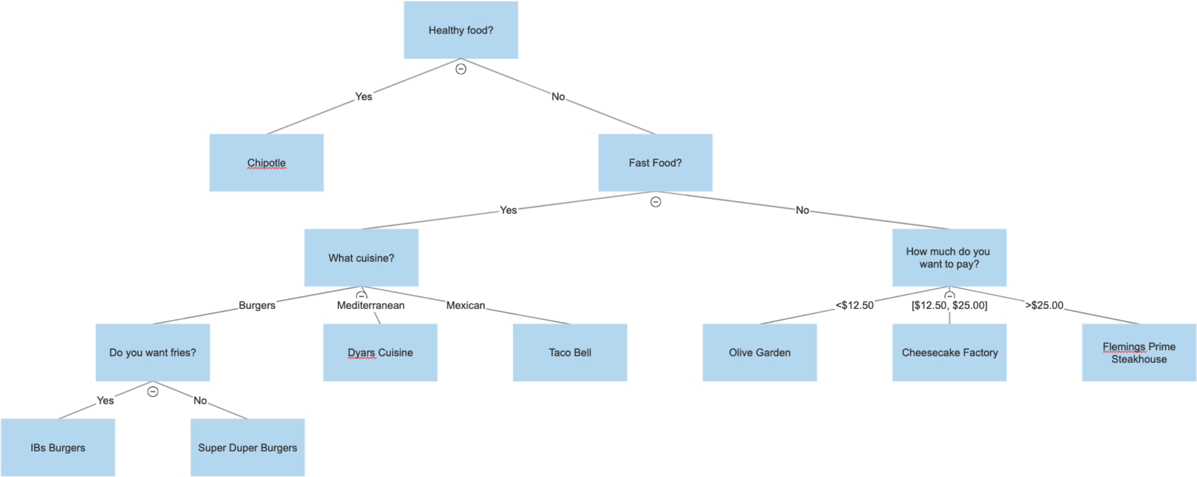
○ $x_2 \geq 5$

○ $x_2 \geq 3.5$

○ $x_2 \geq 6.5$

(d) **(2.0 points)**

We have decided to create a food recommendation system using a decision tree! We would like to run our decision tree to see what food it recommends in certain scenarios.



If you have trouble reading the above tree, please go to this link: https://i.imgur.com/9Z40cYP.png

i. **(1.0 pt)** Bob wants to eat some unhealthy food, specifically at a fast food restaurant. When asked what he's in the mood for, he replies with "Mediterranean". Which of the following restaurants could the decision tree recommend for Bob?

○ Chipotle

○ Taco Bell

● Dyars Cuisine

○ IBs Burgers

ii. **(1.0 pt)** Larry would like to eat some unhealthy food as well! However, he got a salary bonus from his job so he does not want to eat at a fast food restaurant. When asked how much he would like to pay, he replies with "I have no preference". Which of the following restaurants could the decision tree recommend for Larry?

■ Olive Garden

■ Cheesecake Factory

□ Super Dupers Burger

■ Flemings Prime Steakhouse

(e) **(3.0 pt)** Joey and Andrew are each training their own decision tree for a classification task. Joey decides to limit the depth of his decision tree to depth 3 while Andrew decides to not set a limit on the depth of his decision tree. When plotting the training error, Joey's error seems to be much higher than Andrew's error. However, when plotting the validation error, Andrew's error seems be much higher than his training error as well as Joey's error. Andrew is confused and surmises that there must be a bug in his code that is causing this to happen. What happened? Explain. What can he do to improve it? Name at least 3 things he can do to improve his error. Please limit your repsonse to 2 sentences per reason.

He is not correct. Andrew's high validation error and low training error is due to overfitting. Joey did not run into this error because he limited his depth to 3. For Andrew to improve his validation error, he should try to limit the depth of his tree, try pruning his decision tree, preventing splits that have less than 1% of the samples, or using Random forests.

4. (16.0 points)

(a) (3.0 pt) Suppose we are modeling the number of calls to MangoBot food delivery service per minute. We believe that there are likely more calls around lunch time.

Which of the following feature encodings of the time of day (0.0 to 24.0, exclusive of both ends) would capture this assumption? Select all that apply.

□ `time_of_day ** 2`

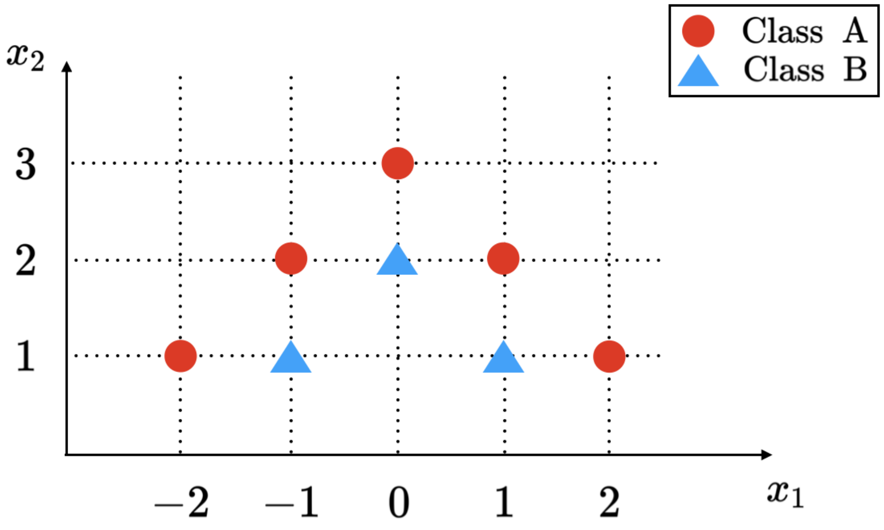
□ `np.log(12 * time_of_day)`

■ `1-np.cos(np.pi * time_of_day / 12)`

■ `np.exp(-(time_of_day - 12) ** 2)`

(b) (4.0 pt) Recall that in a binary classification task, we want our data to become linearly separable so that we can maximize the performance of our classifier. In many cases, however, our data are not directly linearly separable. As a result, we want to apply some transformation to our data so they will become linearly separable afterwards.

For the following dataset, select all transformations that can make the data linearly separable.



■ $(x_1, x_2) \to (x_1^2, x_2)$

□ $(x_1, x_2) \to (x_1, x_2^2)$

■ $(x_1, x_2) \to (x_1^2, x_2^2)$

■ $(x_1, x_2) \to (x_2^2, x_1^2)$

(c) **(3.0 pt)** One way to transform textual data into features is to count the frequencies for all of the words in the text.

Consider the following preprocessing steps:

i. Remove all punctuations (`.`, `,`, `:`, . . .).
ii. Remove all stopwords (`did`, `the`, . . .). Note that stopwords do not include words that negate things such as `no`, `not`, . . .
iii. Lower case the sentence, and keep words that only consist of letters $a - z$.
iv. Encode the sentence as a vector containing the frequencies for all the unique words in the text.

Suppose we use the frequency vector from the steps above as our feature to train a logistic regression model that predicts the sentiment of a sentence (positive, negative). In 1-2 sentences, describe a case where our model would fail and make a false prediction.

Your answer must be specific to the preprocessing steps and includes an example sentence to earn credits.

Counting the frequency of all words in a sentence does not address the order of the words in the sentence. This could be problematic when you have the following two sentences:
"I am happy that it does not rain today." "I am not happy that it does rain today."
The sentiments of the 2 sentences above are clearly opposed to each other, however, if we count the frequency of the words following the same preprocesing steps above, we would end up with exactly the same frequency vector. This means we will always have a false prediction for one of the sentences.

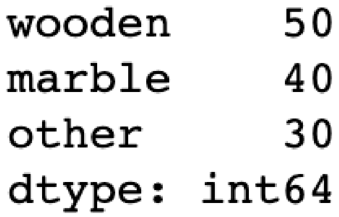
(d) **(3.0 pt)** Recall that in the housing assignment, if we want to include a categorical variable in our linear model, we need to convert it into a collection of dummy variables of values 0 and 1. Suppose we have a dataframe `housing` that contains a subset of the Cook County data.

We are interested in one-hot-encoding the categorical variable `floor_material` and using the dummy columns as the sole features to build an ordinary least squares model to predict the sale price of the houses.

Specifically, we create the design matrix X with the following block of code:

```
X = pd.get_dummies(housing['floor_material']).to_numpy()
```

In addition, running the code `housing['floor_material'].value_counts()` gives us the following output:



```
wooden     50
marble     40
other      30
dtype: int64
```

Which of the following statements are true about the design matrix X? Select all that apply. Note: define θ^* to be the vector containing the optimal parameters.

■ X has a dimension of 3 columns and 120 rows.

□ We can add a bias column of all 1's to X and still find a unique solution for the optimal parameters.

■ $X^\top X$ is a diagonal matrix (zeros everywhere except along the main diagonal).

■ All of the entries in $X^\top X$ add up to be 120.

■ The optimal parameter vector θ^* contains the average sale price for each type of floor material.

(e) **(3.0 pt)** When building your models, one way to select features is to consider the pair-wise relationship between each column and the response variable (i.e. the column you are trying to predict). Consider the following approach:

i. Compute the pairwise correlation coefficient between each column and the response variable in the dataframe.
ii. Sort the correlation coefficients in descending order.
iii. Pick the top k coefficients and select the corresponding columns as the features.

In 1-2 sentences, describe how the approach above can result in multicollinearity and issues with feature diversity.

Your answer must explain why multicollinearity and lack of feature diversity could potentially occur to earn credits for this question.

It is possible that more than 1 column can share a strong correlation with the response variable at the same time, and even worse, there can be strong correlations (near multicollinearity) among multiple columns – this could cause a high variance in the model and impact test performance and is unfortunately not captured by the approach above.
The approach above is deterministic given the data and will always produce a fixed set of columns. This can limit feature diversity as we are building our model.

5. (9.0 points)

Suppose we are modelling some response using our data $\mathbf{X}$. For a given observation we have 3 features, x_1, x_2, x_3. Note that the subscript does **not** refer to the first, second, and third observations, respectively. For a given data point x, we come up with a model of the form $f_\theta(x) = \theta_1 x_1 + \theta_1\theta_2 x_2 + \theta_1^2 x_3$. We use the squared error function, denoted $L(y, \hat{y})$, to calculate the error for each observation and additionally use L2 regularization, denoted $R(\theta)$, with penalty λ. You may assume that $\lambda > 0$. Thus our objective function is of the form $L(y, \hat{y}) + \lambda R(\theta)$.

(a) (3.0 pt) For a single observation x having response y and features x_1, x_2, x_3, compute the gradient to be used in gradient descent:

●
$$-2\begin{bmatrix}(y-(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3))(x_1+\theta_2x_2+2\theta_1x_3)-\lambda\theta_1\\(y-(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3))(\theta_1x_2)-\lambda\theta_2\end{bmatrix}$$

○
$$\begin{bmatrix}2(x_1+\theta_2x_2+2\theta_1x_3-y)(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3)+2\lambda\theta_1\\2(\theta_1x_2-y)(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3)+2\lambda\theta_2\end{bmatrix}$$

○
$$\begin{bmatrix}-2(y-(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3))(x_1+\theta_2x_2+2\theta_1x_3)\\-2(y-(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3))(\theta_1x_2)\end{bmatrix}$$

○
$$2\begin{bmatrix}(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3-y)(\theta_1)+\lambda\theta_1\\(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3-y)(\theta_1\theta_2)+\lambda\theta_1\theta_2\\(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3-y)(\theta_1^2)+\lambda\theta_1^2\end{bmatrix}$$

○
$$\begin{bmatrix}(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3-y)^2(\theta_1)\\(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3-y)^2(\theta_2)\\(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3-y)^2(\theta_3)\end{bmatrix}$$

○
$$\begin{bmatrix}(y-(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3))+\lambda R(\theta_1)\\(y-(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3))+\lambda R(\theta_2)\\(y-(\theta_1x_1+\theta_1\theta_2x_2+\theta_1^2x_3))\end{bmatrix}$$

(b) (2.0 pt) Suppose that you and your friend are implementing gradient descent. Just for fun, your friend chooses a negative learning rate α and asks you to fix their code. Which of the following expressions will **always** result in the same update as the conventional gradient descent algorithm? You may assume that the gradient ∇ is correctly computed and you do not need to worry about the magnitude of α.

□ $\theta^{(t+1)} = \theta^{(t)} - \alpha\nabla$

■ $\theta^{(t+1)} = \theta^{(t)} + \alpha\nabla$

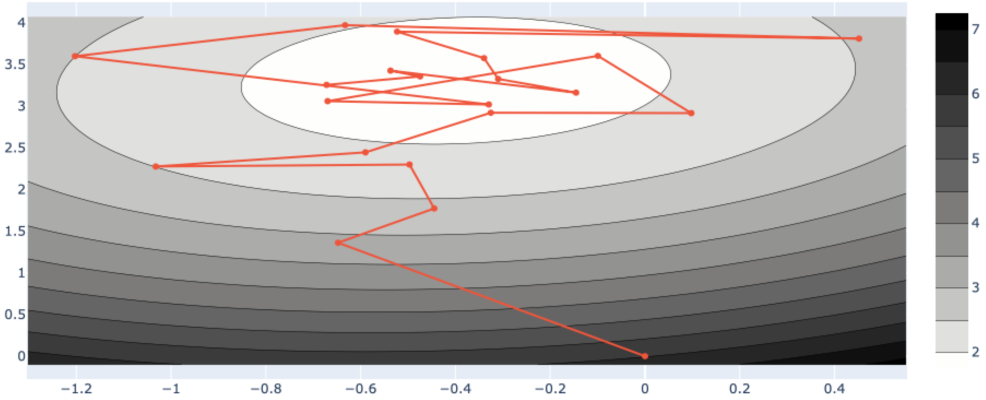
■ $\theta^{(t+1)} = \theta^{(t)} - |\alpha|\nabla$

□ $\theta^{(t+1)} = \theta^{(t)} + |\alpha|\nabla$

□ None of the above

(c) (4.0 points)

i. (2.0 pt) We seek to optimize a given loss function using stochastic gradient descent where $1 <$ batch size $< n$, where n is the total number of data points, We initialize all model parameters as 0 and use a constant learning rate $\eta(t) = \alpha$. Based on the contour plot below, which of the following will most likely result in better minimization of the loss function:



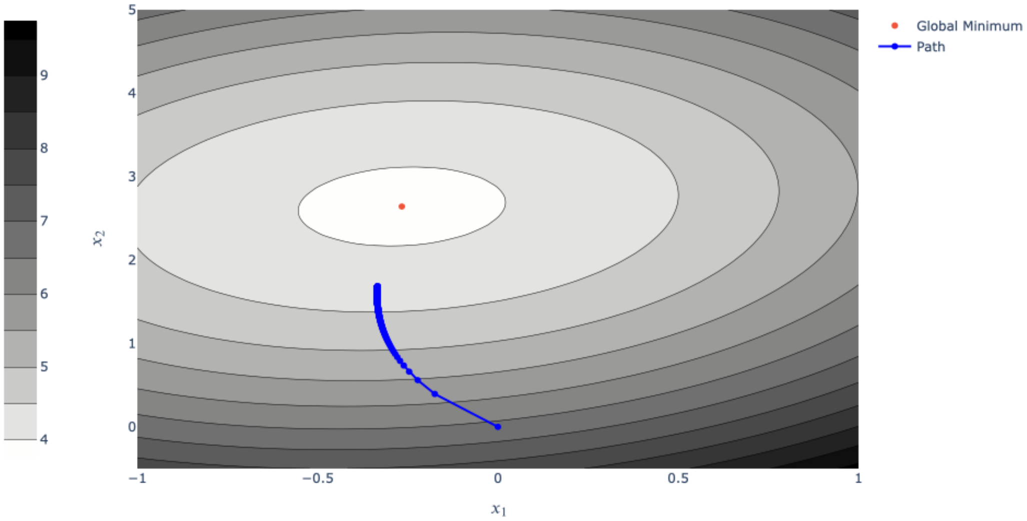
○ Fewer iterations

○ Greater learning rate

○ Smaller batch size

● Greater batch size

ii. (2.0 pt) We seek to optimize a given convex loss function using gradient descent with a decaying learning rate where at time t, the learning rate $\eta(t) = \frac{\alpha}{t+1}$, where $\alpha > 0$. Based on the contour plot below, which of the following will most likely result in better minimization of the loss function:



□ Fewer iterations

■ Greater iterations

□ Negate α

□ Smaller α

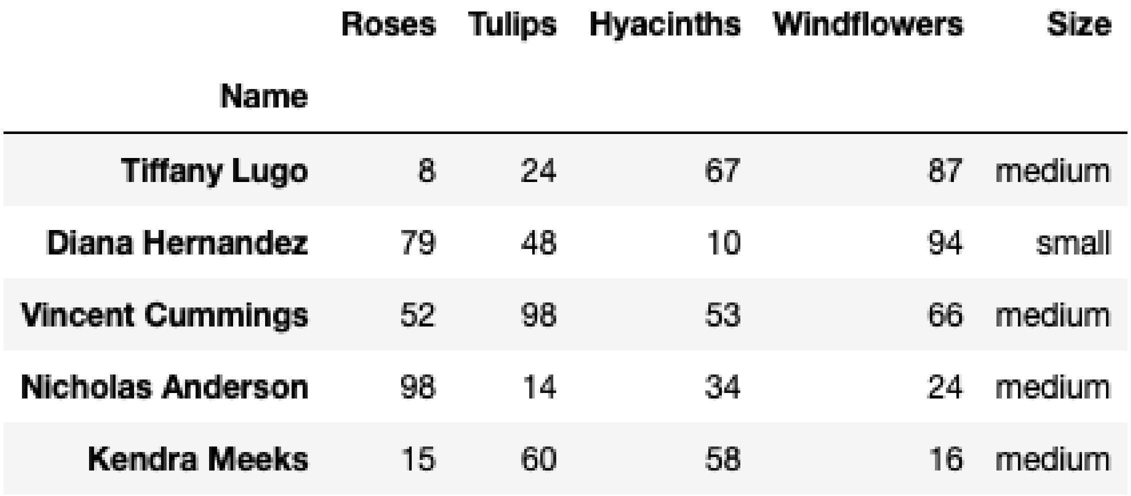
■ Greater α

■ $\eta(t) = \frac{\alpha}{\sqrt{t+1}}$

□ $\eta(t) = \frac{\alpha}{(t+1)^2}$

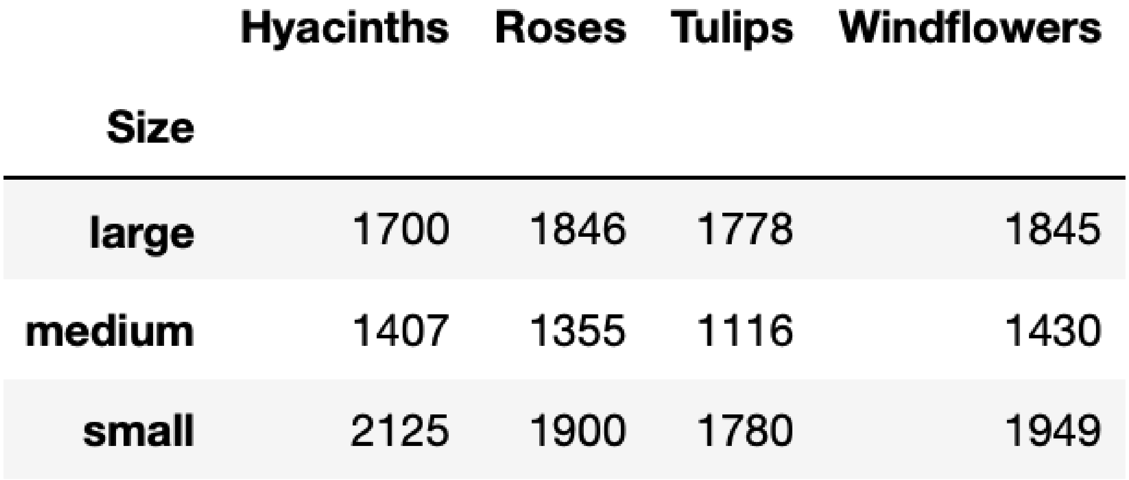
6. **(6.0 points)**

   (a) **(6.0 pt)** Leif wants to do a study on the number of flowers in people's gardens. He collects data on 100 different gardens, classifying each of them into three different sizes: 'small', 'medium', and 'large', and counts every flower in each person's garden. The following is the first five rows of the data he collected:



	Roses	Tulips	Hyacinths	Windflowers	Size
Name					
Tiffany Lugo	8	24	67	87	medium
Diana Hernandez	79	48	10	94	small
Vincent Cummings	52	98	53	66	medium
Nicholas Anderson	98	14	34	24	medium
Kendra Meeks	15	60	58	16	medium

Leif then asks you to construct the following table using the data he collected. The table represents the total flowers in each category. For example, there are 1700 Hyacinths in "large" gardens.



	Hyacinths	Roses	Tulips	Windflowers
Size				
large	1700	1846	1778	1845
medium	1407	1355	1116	1430
small	2125	1900	1780	1949

Write code below such that the above table is generated. Assume the data Leif collected is placed in a Pandas DataFrame assigned to the variable `inputdf`. The resulting table should be named `outputdf`. Please follow the template below (you must use `pd.pivot_table`).

```
outputdf = pd.pivot_table(_________________________________________________)
```

```
pivot_df = pd.pivot_table(data=new_df, values=new_df.columns[:4], index='Size', aggfunc='sum')
```

7. (8.0 points)

(a) (8.0 pt) Kunal has a large dataset of Irish poems. He wants to analyze the many different sentences in each of the poems. He has a list of words he is particularly interested in:

```
words = ['artist', 'dinner', 'data', 'pay', 'color', 'science', 'clearly', 'run']
```

Kunal creates a Pandas Series of 100 sentences from these poems and wants to build a frequency array for the above words for each of the 100 sentences. The frequency array should capture how many times a word in the above list of words appears in a certain sentence.

For example, the sentence "Data Science is clearly science." would yield an array like:

```
[0, 0, 1, 0, 0, 2, 1, 0]
```

Note: 'science' was recorded twice, even though the first letter has different capitalization.

You may assume all collected sentences have no punctuation.

Define a function that takes in a Series of sentences and a list of words as an input, and outputs a frequency DataFrame where each row represents a frequency array for each sentence, as described above. Please start your code with the following method signature:

```
def funcname(ser, words):
```

Note: Please limit your response to 6 lines. The staff's solution was done in 3 lines, for reference (including the function signature). Hint: Try using one of the following str methods: `str.contains`, `str.get`, `str.count`, `str.split`, `str.find`.

```
def f(ser, words):
    freq_array = np.array([ser.str.lower().str.count(word) for word in words]).T
    return pd.DataFrame(freq_array, columns=words)
```

8. (8.0 points)

(a) (3.0 pt) Suppose I am given a dataset [-2,4,1,3,4] and I am using the constant model, $\hat{y_i} = \theta$. To find the optimal θ^*, I have the following two loss functions to work with. $L_A(y_i, \theta) = (3y_i - \theta)^2$ and $L_B(y_i, \theta) = |2y_i - \theta|$. Let θ_A^*, θ_B^* be the optimal parameters found for loss functions L_A, L_B respectively. What is the relationship between θ_A^* and θ_B^*?

○ $\theta_A^* > \theta_B^*$

● $\theta_A^* = \theta_B^*$

○ $\theta_A^* < \theta_B^*$

○ Need more information to tell

(b) (2.0 pt) Which of the following models is linear in the parameters?

■ $f_\theta(x) = \theta_1 x + \theta_2 x^2 + \theta_3 e^{cos(x)}$

□ $f_\theta(x) = sin(\theta_1)x + \theta_2 x$

■ $f_\theta(x) = \theta_1$

□ $f_\theta(x) = \theta_1 log(x^4 + 5x^3 + 6) + \theta_2^3 x$

■ $f_\theta(x) = \frac{1}{x^2+1}\theta_1 + \theta_2$

(c) (3.0 pt) Which of the following is true regarding MSE (Mean Squared Error) & MAE (Mean Absolute Error) in Linear Regression?

□ There is a closed form solution to the optimal parameters when using MAE.

■ If our data contains many corrupted outliers, MAE loss is a better metric than MSE loss.

■ MAE encourages sparsity in the parameters (a lot of parameters are set to 0), which allows for non-relevant features to not be included.

□ The median minimizes the MSE for a constant model.

□ The optimal parameters found in a MSE loss function and a MAE loss function will never be equal.

■ The MAE loss is not differentiable everywhere, which makes it impossible to take the gradient at those points.

9. (16.0 points)

In this question we will be focusing on predicting whether a tweet is happy (1) or sad (0) using logistic regression. Rather than using a bag-of-words featurization, we will simply count the number of positive emojis (`":-)"`, `";-)"`, ...) and negative emojis (`":-["`, `":-("`, ...).

Assume you are given a training dataframe `training` of the following form (the first 4 rows are shown):

Tweet	HappyEmojiCount	SadEmojiCount	isHappy
Woke up to a sunny day :-). Life is good :)	2	0	1
Stuck in traffic for 1 hr on my way to work today (._.)	0	1	0
Found a new album that really slaps =^_^= check it out on my Spotify	1	0	1
Grinding on this paper until 2am last night (-_-), but last paper of the semester :)	1	1	1

You fit a logistic regression model using the following block of codes:

```
from sklearn.linear_models import LogisticRegression

lr = LogisticRegression(intercept=True)
lr.fit(training[[‘‘HappyEmojiCount’’, ‘‘SadEmojiCount’’]], training[‘‘isHappy’’])
```

(a) **(3.0 pt)** Given a new tweet transformed into a numpy array containing the same set of features, and the array is assigned to the variable x_{test}, which of the following expressions computes: $P(\text{Sad } | x_{\texttt{test}}) = P(Y = 0|X = x_{\text{test}})$ under the logistic regression model (note the label Sad is the same as Not isHappy here)?

Note: θ is the vector containing the trained parameters from the logistic regression model. σ is the sigmoid function. $\mathbf{1}\{x\}$ is a function that returns 1 if x is true and 0 otherwise.

○ $\sigma(\theta^\top x_{\texttt{test}})$.

● $1 - \sigma(\theta^\top x_{\texttt{test}})$

○ $\sigma(1 - \theta^\top x_{\texttt{test}})$

○ $\mathbf{1}\{\sigma(\theta^\top x_{\texttt{test}}) > 0.5\}$

○ $\mathbf{1}\{\sigma(\theta^\top x_{\texttt{test}}) < 0.5\}$

(b) **(4.0 pt)** Using the same model, we are now interested in seeing **how much more likely** our model will classify a tweet as happy rather than sad. We will use the following metric (note the log here is in natural base):

$$\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right)$$

Suppose we have the following tweet and the corresponding features:

Tweet	HappyEmojiCount	SadEmojiCount
Weather is a bit dry today (=_=) :(stay hydrated during your workout () :) :-)	3	2

Our model has the following set of trained parameters:

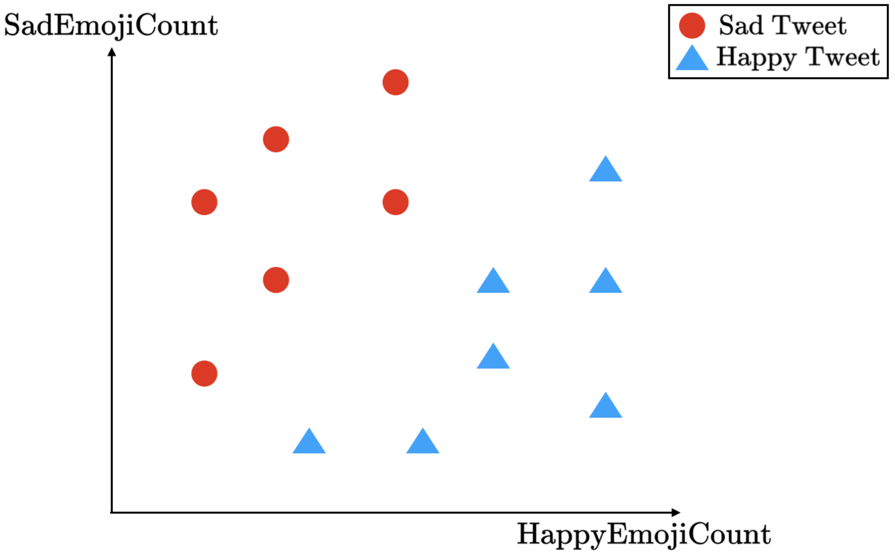
Features	Intercept Term	HappyEmojiCount	SadEmojiCount
Coefficient	0.2	0.7	-0.5

What is the value of the metric for this tweet?

(c) (4.0 points)

Consider a small subset of tweets scattered as points (HappyEmojiCount, SadEmojiCount) in the 2-dimensional plane. We will use the shape and color of a point to indicate whether the tweet is actually happy or sad.

Suppose for the following subset of tweets, we want to train a logistic regression model (intercept included) with $L2$ regularization on one of its parameters.



Recall that a logistic regression model with $L2$ regularization has the loss function of the following form:

$$\text{Loss}(Y, \hat{Y}, \theta) = \text{CrossEntropyLoss}(Y, \hat{Y}) + \lambda\theta_i^2,$$

where θ_i is some parameter from the model.

Consider the figures below depicting three possible decision boundaries applied on the dataset.

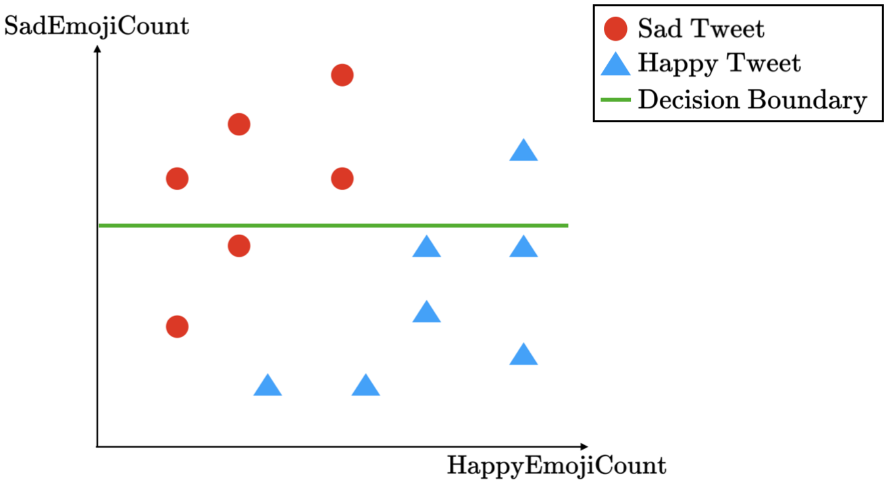


Figure A

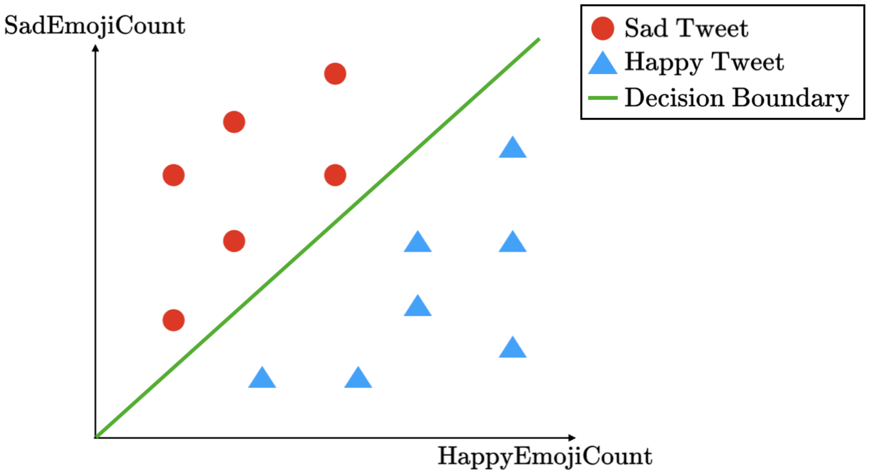


Figure B

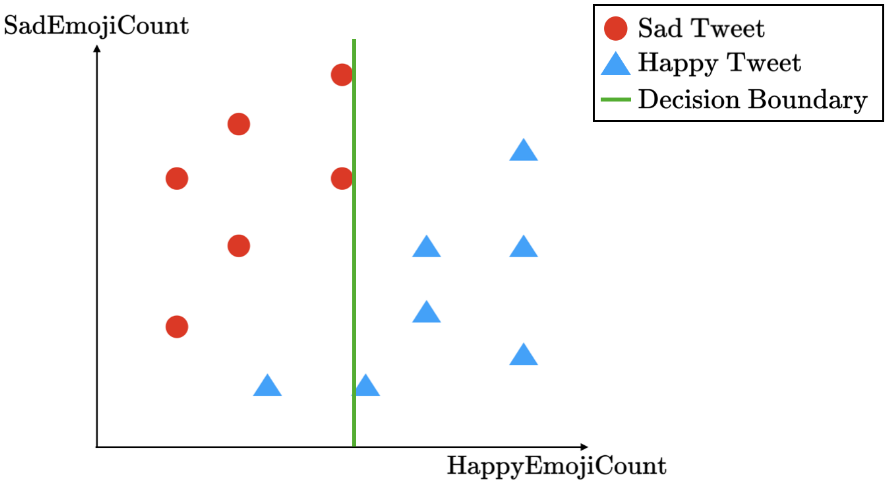


Figure C

For each of the following parameters, explain, by matching with one of the figures above, when compared to a model with no regularization, what would happen to the slope of the decision boundary that divides the two classes and the training error if we set λ to be a really large value.

i. (1.0 pt) Let θ_i = coefficient for the sad emoji count. Which of the figures above best depict the decision boundary in this case?

○ Figure A

○ Figure B

● Figure C

○ None of the above

ii. (1.0 pt) Let θ_i = coefficient for the sad emoji count. What will happen to the training error in this case?

● Increase

○ Decrease

○ Remain the same

○ Cannot be determined

iii. (1.0 pt) Let θ_i = coefficient for the happy emoji count. Which of the figures above best depict the decision boundary in this case?

● Figure A

○ Fgure B

○ Figure C

○ None of the above

iv. (1.0 pt) Let θ_i = coefficient for the happy emoji count. What will happen to the training error in this case?

● Increase

○ Decrease

○ Remain the same

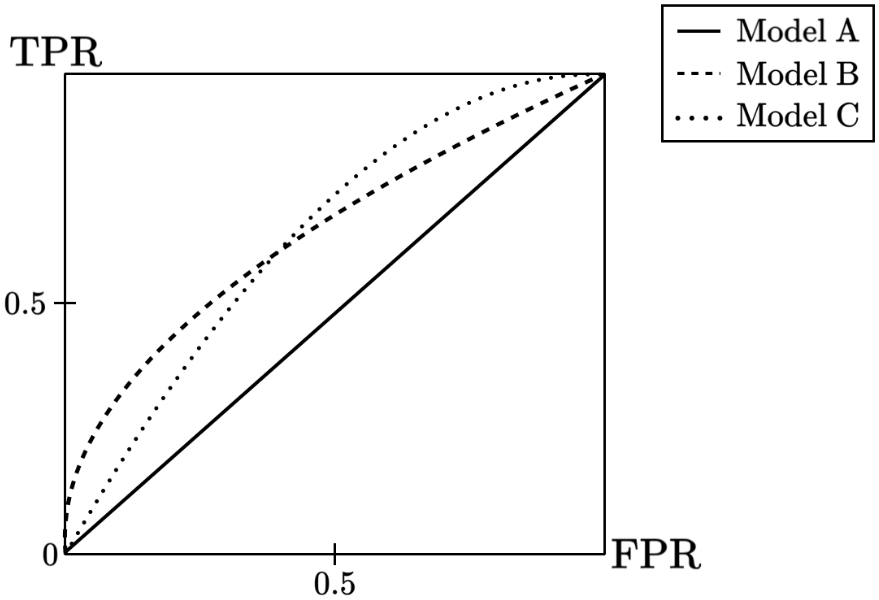
○ Cannot be determined

(d) (2.0 pt) We test our logistic regression model on a subset of tweets with 70-30 class imbalance. In other words, 70% of the tweets are happy and the remaining 30% are sad. Unfortuntely, it turns out our model only has a 40% accuracy. Suppose we invert the predictions from our model, i.e. if a model predicts happy, output sad instead, and vice versa. **In percentages**, what would be the new accuracy of our model?

Please round your answer to the nearest integer between 0 and 100.

(e) (3.0 points)

Consider the following 3 logistic regression models with different features trained on the same dataset. Let the models be denoted A, B, and C respectively. Shown below are the corresponding ROC curves for these models:



i. (2.0 pt) Suppose we fix the decision threshold for all 3 logistic regression models to be such that we get a FPR of 0.8 when we evaluate our model. In order of most to least preferred, rank each of the models given their ROC curves.

● $C > B > A$

○ $C > A > B$

○ $B > C > A$

○ $B > A > C$

○ Cannot be determined

ii. (1.0 pt) Which of the following model is a strictly worse classifer?

● A

○ B

○ C

○ Cannot be determined

10. (7.0 points)

Let us say that we have some model $y_1 = \theta_1 X$ (notice no intercept term), where $\theta_1 \in \mathbb{R}$ and $X, y_1 \in \mathbb{R}^{n \times 1}$ (vectors), which takes in n data points with the same common y-value c. In other words, the model is fitted to the data points $(x_1, c), (x_2, c) \ldots (x_n, c)$.

Let us also say that we have some model $y_2 = \theta_2 X$ (notice no intercept term), where $\theta_2 \in \mathbb{R}$ and $X, y_2 \in \mathbb{R}^{n \times 1}$ (vectors), which takes in the same x-values of the n data points, **except** with the common y-value c'. In other words, the model is fitted to the data points $(x_1, c'), (x_2, c') \ldots (x_n, c')$.

(a) (3.0 pt) If we run OLS on our data, what will be the value of θ_1? Note that in the answer choices, $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

● c

○ $c \cdot \left(\frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} x_i}\right)$

○ $c \cdot \left(\frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2}\right)$

○ $c \cdot \left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)$

○ None of the above

(b) (4.0 pt) We now choose to combine all our data points and fit to a combined model:

$$y_{\text{comb}} = \theta_{\text{comb}} X_{\text{comb}}$$

Where $\theta_{\text{comb}} \in \mathbb{R}$ and $X_{\text{comb}}, y_{\text{comb}} \in \mathbb{R}^{2n \times 1}$. In other words, the model is fitted to the data points $(x_1, c) \ldots (x_n, c), (x_1, c'), \ldots, (x_n, c')$ and are fed into y_{comb}.

Which of the following is the relationship between $\theta_{\text{comb}}, \theta_1$, and θ_2?

● $\theta_{\text{comb}} = \frac{\theta_1 + \theta_2}{2}$

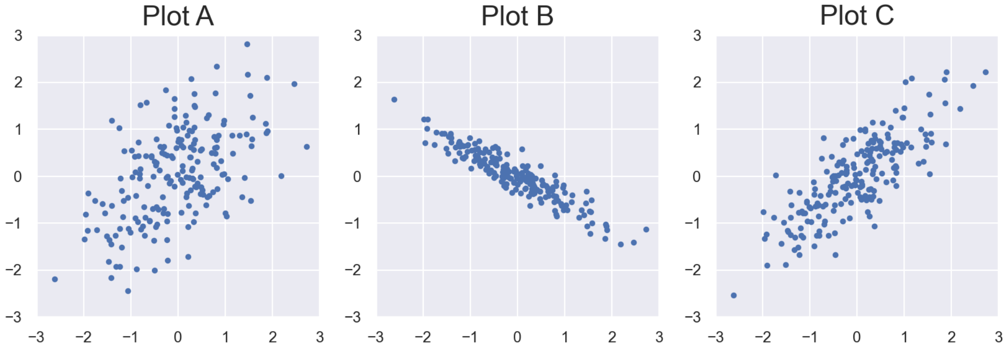
○ $\theta_{\text{comb}} = \frac{c\theta_1 + c'\theta_2}{c + c'}$

○ $\theta_{\text{comb}} = c\left(\frac{\theta_1}{\sum_{i=1}^{n} x_i}\right) + c'\left(\frac{\theta_2}{\sum_{i=1}^{n} x_i}\right)$

○ None of the above

11. (6.0 points)

Suppose we are given three datasets A, B, and C $\in \mathbb{R}^{100\times 2}$ i.e. each dataset consists of 100 data points in two dimensions. We visualize the datasets using scatterplots, labelled Plot A, Plot B, and Plot C, respectively:



(a) **(1.0 pt)** If we applied PCA to each of the above datasets and used only the first principal component which dataset(s) would have the lowest reconstruction error?

□ Dataset A

■ Dataset B

□ Dataset C

□ Cannot be determined

(b) **(2.0 pt)** If we applied PCA to each of the above datasets and used the first two principal components, which dataset(s) would have the lowest reconstruction error?

■ Dataset A

■ Dataset B

■ Dataset C

□ Cannot be determined

(c) **(3.0 pt)** Suppose we are taking the SVD of one of the three datasets, which we will name dataset X. We run the the following piece of code:

```
X_bar = X - np.mean(X, axis=0)
U, Sigma, V_T = np.linalg.svd(X_bar)
```

We get the following output for `Sigma`:

```
array([15.59204498,  3.85871854])
```

and the following output for `V_T`:

```
array([[ 0.89238775, -0.45126944],
       [ 0.45126944,  0.89238775]])
```

Based on the given plots and the SVD, which of the following datasets does dataset X most closely resemble:

○ Dataset A

● Dataset B

○ Dataset C

12. (6.0 points)

Suppose you are estimating a true function $g(z) = Az^2$ for $z \in \mathbb{R}$ (i.e. z is a scalar) with **ordinary least squares linear regression**, where the model is $f_\theta(z) = \theta z$ and $\theta \in \mathbb{R}$. We train the model with just one training data point x, y, generated according to $Y = g(x) + \epsilon$. Assume $\epsilon \sim \mathcal{N}(0, \sigma^2)$ (i.e. ϵ normally distributed with mean 0 and variance σ^2).

For the rest of the question, assume $x = 1$.

(a) **(2.0 pt)** For this part of the question, assume $\sigma^2 = 0$. In other words, **there is no noise in the labels.** What is the bias and variance of the model f_θ at a test point $z = 1$?

Hint: Start by solving for the optimal choice of θ given x.

● bias=0, variance=0

○ bias=A, variance=σ^2

○ bias=A, variance=0

○ bias=0, variance=σ^2

Since the test point $z = 1$ is the same as the single data point we used for training ($x = 1$), we can expect an unbiased prediction. Since there is no noise in the labels, there will be no variance to our prediction since the label $g(x)$ used for training will be the same no matter how many times we repeat this process.

Alternatively, you can work out the math.

The optimal choice for θ (which we refer to as θ^*), can be obtained in 2 ways:

Method 1: Normal Equations Apply the normal equations with the matrix X being the scalar x and $y = Ax^2$ is the label for x.

$$\theta^* = (X^T X)^{-1} X^T y \ \theta^* \quad = (x \cdot x)^{-1} \cdot x \cdot Ax^2 \ \theta^* = Ax$$

Method 2: Directly minimize MSE loss The loss function used in least squares is MSE. We can use derivatives to solve for θ^*.

$$\theta^* = \min_\theta MSE(\theta)$$

where $MSE(\theta) = (f_\theta(x) - y)^2$.

Plugging in $f_\theta(x)$ and using chain rule, we get:

$$\frac{d}{d\theta} MSE(\theta) = 2(\theta x - y) \cdot x$$

Setting $\frac{d}{d\theta} MSE(\theta) = 0$ and solving for θ gives:

$$\theta^* = Ax$$

Using θ^*, we can write our trained model as $f_{\theta^*}(x) = Axz = Az$.

From here, you can see that the given plot

We compute bias at $z = 1$ using the definition of bias:

$$bias(z) = \mathbb{E}[f_{\theta^*}(z) - g(z)] \ bias(z) \quad = \mathbb{E}[f_{\theta^*}(z)] - g(z) \ bias(z) = Az - Az^2 \ bias(z) \quad = -A(z - z^2) \ bias(z = 1) = -A($$

where we observe $\mathbb{E}[f_{\theta^*}(z)] = Az$ because the expectation is just over our one point x.

We similarly compute variance:

$$variance(z) = \mathbb{E}[(f_{\theta^*}(z) - g(z))^2] \ variance(z) \quad = \mathbb{E}[(f_{\theta^*}(z)^2] - 2g(z)\mathbb{E}[(f_{\theta^*}(z)] + g(z)^2 \ variance(z) = (Az)^2 - 2Az^2 A$$

(b) **(4.0 pt)** For this part of the question, let $\sigma^2 > 0$. Select the correct statement about the bias and variance of the model f_θ at a test point $z = 1$.

○ bias=0, variance=0

○ bias=A, variance=σ^2

○ bias=A, variance=0

● bias=0, variance=σ^2

Since on average the noise is 0, we can still expect an unbiased prediction.

Working out the math: Using the same approach as the previous part, you can obtain

$$\theta^* = Ax + \frac{\epsilon}{x} = A + \epsilon$$

since $x = 1$. The trained model is then $f_{\theta^*}(x) = (A + \epsilon)z$.

Computing bias:

$$bias(z) = \mathbb{E}[f_{\theta^*}(z) - g(z)]\ bias(z) \quad = \mathbb{E}[f_{\theta^*}(z)] - g(z)\ bias(z) = \mathbb{E}[(A+\epsilon)z] - g(z)\ bias(z) \quad = Az - Az^2\ bias(z) = -$$

where $\mathbb{E}[(A+\epsilon)z] = Az$ because $\mathbb{E}[\epsilon] = 0$.

The noise in the labels shows up in the variance of our prediction.

$$variance(z) = \mathbb{E}[(f_{\theta^*}(z) - g(z))^2]\ variance(z) \quad = \mathbb{E}[(f_{\theta^*}(z)^2] - 2g(z)\mathbb{E}[(f_{\theta^*}(z)] + g(z)^2\ variance(z) = (Az)^2 + \sigma^2 -$$

where $\mathbb{E}[(f_{\theta^*}(z)^2]$ is expanded as:

$$\mathbb{E}[(f_{\theta^*}(z)^2] = \mathbb{E}[((A+\epsilon)z)^2] \quad = \mathbb{E}[(Az)^2 + 2Az\epsilon z + (\epsilon z)^2] \ = (Az)^2 + 2Az\mathbb{E}[\epsilon z] + \mathbb{E}[(\epsilon z)^2] \quad = (Az)^2 + 0 + z^2\mathbb{E}[\epsilon^2] \ =$$

13. (5.0 points)

(a) **(1.0 pt)** Alice is training a model and finds that as she adds more features her training error is decreasing along with her validation error. What should she do?

○ Increase regularization.

○ Decrease regularization.

● No additional regularization changes are needed.

(b) **(1.0 pt)** Alice is training a model and her training error is rapidly decreasing but her validation error is increasing. What should she do?

● Increase regularization.

○ Decrease regularization.

○ No additional regularization changes are needed.

(c) **(1.0 pt)** Suppose you are interested in finding the minimal set of explanatory features, which form of regularization would be most appropriate?

● L1 regularization.

○ L2 regularization.

○ No regularization.

(d) **(1.0 pt)** Suppose Alice finds that her model is overfitting and she decides to add L2 regularization with regularization coefficient λ to her model. As she increases the regularization coefficient λ, which of the following are true?

■ Bias increases

□ Bias decreases

□ Variance increases

■ Variance decreases

(e) **(1.0 pt)** Consider a simple setting in which we are predicting the height of a person in centimeters based on their weight. Suppose we include the weight measured in kilograms (kg) and milligrams (mg) as two separate features and we tune the coefficient of the L1 regularization to include only one feature. Without normalizing the data before training, which feature would be selected after the model is trained?

● Weight in mg

○ Weight in kg

14. (8.0 points)

There are 10 blue, 15 red, and 25 green balls in a bag from which we sample uniformly at random without replacement. Let I_i be the indicator that the i-th ball drawn will be red. Calculate the following terms:

(a) (1.0 pt) $E[I_1]$

0.3

(b) (1.0 pt) $Var[I\_1]$

0.21

(c) (2.0 pt) $E[I\_1 + I\_50]$

0.6

(d) (2.0 pt) $E[\sum\_i = 1^{50} I\_i]$

15

(e) (2.0 pt) $Var[\sum\_i = 1^{50} I\_i]$

0

No more questions.